Generalized Planning: Non-Deterministic Abstractions and Trajectory Constraints

B. Bonet\textsuperscript{1}  G. De Giacomo\textsuperscript{2}  H. Geffner\textsuperscript{3}  S. Rubin\textsuperscript{4}

\textsuperscript{1} Universidad Simón Bolívar, Venezuela
\textsuperscript{2} Sapienza Università di Roma, Italy
\textsuperscript{3} ICREA & Universitat Pompeu Fabra, Spain
\textsuperscript{4} Università degli Studi di Napoli Federico II, Italy
Generalized Planning: Example

Want policy that works for many (possibly $\infty$) problem instances

Example: Problem Counter-$n$:
- Counter problem with **single variable** $X$ with initial value $X = n$
- Agent **senses** whether $X = 0$ or $X > 0$
- Agent can **increase** or **decrease** value of $X$
- **Observable goal** is to reach $X = 0$

Policy: “if $X > 0$, decrease $X$” works:
- for any $n \geq 0$
- problems with more than one possible initial state
- even if actions may fail sometimes; e.g. decrease don’t work sometimes

(Srivastava et al. 2008, 2011; Hu & Levesque 2010; Hu & De Giacomo 2011; B. & Geffner 2015; Belle & Levesque 2016; etc)
Generalized Planning: Formulation

- In Hu & De Giacomo (2011) formulation, collection $\mathcal{P}$ of instances assumed to share common pool of observations and actions

- Policy $\mu$ mapping observations into actions said to generalize to $\mathcal{P}$ if it solves all problems in $\mathcal{P}$

- General finite-state controllers can be defined in same way
Generalized Planning: Computation

Top-down approach:

- If $\mathcal{P}$ is finite, **compile** $\mathcal{P}$ into a regular planning problem or do **search** in controller space (Hu & De Giacomo 2011)

- If $\mathcal{P}$ is infinite, finite subset of $\mathcal{P}$ sometimes ensures generalization to $\mathcal{P}$ (e.g. 1D problems; Hu & Levesque 2010)

Bottom-up approach:

- Solve **single** “representative instance” $P$ of $\mathcal{P}$ and **prove** that solution ensures generalization (B. et al. 2009)

- **Example:** solution to Counter problem with **two** possible initial states **generalizes** to class $\mathcal{P} = \{\text{Counter-}n : n \geq 0\}$
Goal for this paper

Key question in **bottom-up approach:**

– What’s the **common structure** between **single problem** $P$ and class $\mathcal{P}$ that yields the generalization?

Question partially answered in earlier work:

**Theorem (B. & Geffner 2015)**

> If $P$ reduces to $P'$ and $\mu$ is strong cyclic solution for $P'$, then $\mu$ solves $P$ if it terminates in $P$ over fair trajectories

In this work, we:

– analyze necessity of **termination** in B. & Geffner (2015) formulation
– show how to **get rid** of termination condition
Outline

• Basic framework
• Observation projections abstractions
• Trajectory constraints
• New generalization theorems
• Generalized planning as LTL Synthesis
• Generalized planning over QNPs as FOND planning
• Wrap up
Partially obs. non-det. problem $P = (S, I, \Omega, Act, T, A, obs, F)$:

- $S$ is state space (finite or infinite)
- $I \subseteq S$ is set of initial states
- $\Omega$ is set of observations
- $Act$ is set of actions
- $T \subseteq S$ is set of goal states
- $A : S \rightarrow 2^{Act}$ is available-actions function
- $obs : S \rightarrow \Omega$ is observation function
- $F : Act \times S \rightarrow 2^S \setminus \{\emptyset\}$ is non-deterministic transition function

Class $\mathcal{P}$ of PONDPs with observable goals and action preconditions, and where all problems share common:

- set of actions $Act$
- set of observations $\Omega$
- subset $T_\Omega$ of goal observations; $\forall P \forall s : s \in T_P \iff obs_P(s) \in T_\Omega$
- subsets $A_\omega$ of actions: $\forall P \forall s : A_P(s) = A_{obs(s)}$
Standard Solution Concepts

Policy is function $\mu : \Omega^+ \rightarrow \text{Act}$

Policy $\mu$ is **valid** for problem $P$ if it selects **applicable actions**

Let $P$ be a problem and $\mu$ be a valid policy for $P$:

- $\mu$ is **(strong) solution** for $P$ iff every $\mu$-trajectory is goal reaching
- $\mu$ is **fair solution** or **strong cyclic solution** for $P$ iff every **fair** $\mu$-trajectory is goal reaching

Henceforth, we focus on valid policies
Abstractions: Observation Projection

Project entire class $\mathcal{P}$ into single non-deterministic problem $P^o$: 

- **state space:** $S^o = \Omega$

- **initial states:** $\omega \in I^o$ iff $\text{obs}_P(s) = \omega$ for some $P$ and $s \in I_P$

- **actions:** $\text{Act}^o = \text{Act}$ and $A^o(\omega) = A_\omega$

- **goal states:** $T^o = T_\Omega$

- **transitions:** $\omega' \in F^o(a, \omega)$ iff $s' \in F_P(a, s)$ for some problem $P$ in $\mathcal{P}$, and states $s$ and $s'$ with $a \in A_P(s)$, $\text{obs}_P(s) = \omega$ and $\text{obs}_P(s') = \omega'$

**Example:** For class of Counter-$n$ problems, $P^o$ features:

- **2 states** (observations): $[X = 0]$ and $[X > 0]$

- **non-deterministic** transitions; e.g. $[X > 0]$ transitions under decrease action to both $[X = 0]$ and $[X > 0]$
Need for More Structure

Policy \( \mu = "\text{if } X > 0, \text{ decrement } X" \) solves all Counter-\( n \) problems but doesn’t solve projection \( P^o \)

\( P^o \) is non-deterministic and \( \mu \) may get trapped into loop where Decrement \( X \) doesn’t work

Projection \( P^o \) misses important structural property that all Counter-\( n \) problems share but that is lost projection:

If variable \( X \) is decreased infinitely often and increased only a finite number of times, it eventually reaches \( X = 0 \)

In this work we extend the model to make such properties explicit
Trajectory Constraints

Trajectory constraint $C$ over $P$ is subset of infinite state-action sequences (i.e. $C \subseteq (S \times Act)^\infty$) or subset of infinite observation-action sequences (i.e. $C \subseteq (\Omega \times Act)^\infty$)

Trajectory $\tau$ satisfies $C$ if $\tau$ is finite, or either $\tau \in C$ (if $C \subseteq (S \times Act)^\infty$), or $obs(\tau)$ in $C$ (if $C \subseteq (\Omega \times Act)^\infty$) where

$$obs(\langle s_0, a_0, s_1, a_1, \ldots \rangle) = \langle obs(s_0), a_0, obs(s_1), a_1, \ldots \rangle$$

- Problem $P$ extended with constraint $C'$ is denoted by $P/C$
- Problem $P$ satisfies constraint $C$ if all trajectories in $P$ satisfy $C'$

New solution concept: $\mu$ solves $P/C$ iff every $\mu$-trajectory $\tau$ that satisfies $C'$ is goal reaching

Example: $C = \{\tau : \tau$ is infinite and satisfies the crucial property$\}$ for $P$
Theorem (Generalization)

Let $\mathcal{P}$ be a class of FOND$P$ and $C$ a constraint such that every $P$ in $\mathcal{P}$ satisfies $C$. Then, $\mu$ solves all problems in $\mathcal{P}$ if $\mu$ solves $P^o/C$.

Example: trajectories in $P^o$ that satisfy $C$ happen to be fair. Thus, $\mu$ must be fair solution ($P^o$ has no strong solution by non-determinism). Theorem asserts $\mu$ solves all instances in which decrease action satisfies constraint.

Theorem (Completeness)

If $P^o$ is obs. projection for class $\mathcal{P}$ and $\mu$ solves all problems in $\mathcal{P}$, there is constraint $C$ over $P^o$ such that every $P$ in $\mathcal{P}$ satisfies $C$ and $\mu$ solves $P^o/C$. 
Generalized Planning as LTL Synthesis

When trajectory constraints can be expressed in LTL (over language $\Sigma = \text{Act} \cup \Omega$), LTL techniques can be used to obtain general plans.

**Theorem**

Let $P^o/C$ be obs. projection with constraint $C$ expressed in LTL as $\Psi$. Then, solving $P^o/C$ (and hence all $P/C$ for $P \in \mathcal{P}$) is 2EXPTIME-complete; it’s double-exponential in $|\Psi| + |T^o|$ and polynomial in $|P^o|$.

**Sketch:** Idea is to think of policies $\mu$ as $\Omega$-branching $\text{Act}$-labeled graph:

- Build **tree-automaton** accepting policies $\mu$ such that every $\mu$-trajectory satisfies formula $\Phi = \Psi \supset \Diamond T^o$ where $\Diamond T^o$ is reachability goal in $\mathcal{P}$

- Check **non-emptiness** of language accepted by tree-automaton; this test yields witness (i.e. policy) if it exists
Qualitative Numerical Planning

- Problem $R_V$ with set $V$ of non-negative numeric variables (don’t have to be integer variables) and standard Boolean propositions
- Actions can affect propositions and also increase or decrease value of numeric variables non-deterministically
- Propositions are fully observable while only $X = 0$ and $X > 0$ can be observed for each var $X$
- Paper describes syntax for specifying class of QNPs sharing same set of vars, fluents, actions, observations, . . .

Example: General problem of stacking a block $x$ on a block $y$ in instance with any number of blocks can be cast as QNP

Abstractions for some QNPs appear in (Srivastava et al., 2011, 2015)
Solving QNPs with FOND Planners

Given QNP $R_V$, obs. projection $R^o_V$ constructed syntactically:

- Projection contains only propositions and no numeric variables
- For each variable $X$, there are propositions $X > 0$ and $X = 0$
- Each effect $Inc(X)$ replaced by atom $X > 0$, and effect $Dec(X)$ replaced by non-det. effect $X > 0 \mid X = 0$

Non-determinism in $P^o$ isn’t fair (Srivastava et al. 2011); i.e. strong cyclic plan for $R^o_V$ isn’t guaranteed to be solution

Projection $R^o_V$ is modified to target interesting subclasses of QNPs:

Theorem (Soundness and Completeness)

Let $R_V$ be QNP such that a) actions with $Dec(X)$ effects have prec. $X > 0$, and b) actions have decrement effects for at most one variable. $\mu$ is fair solution to modified $R^o_V$ iff $\mu$ solves all problems in class defined by $R_V$
Related Work

- QNPs related to problems considered by (Srivastava et al. 2011, 2015)

- 1D problems (Hu & Levesque 2010; Hu & De Giacomo 2011) is infinite class of “identical” problems characterized by single integer parameter

- Hu & De Giacomo (2011) construct a single “large enough” abstraction whose solution provides a solution to the class

- Sardiña et al. (2006) also analyze tasks in which “global properties” are lost in observation projection; we recover such properties with constraints

- De Giacomo et al. (2016) show that trace constraints are necessary for belief construction to work on infinite domains
Summary

- Bottom-up approach for generalized planning where general policies are obtained from solutions of single instances

- Non-deterministic abstraction $P^o$ extended with trajectory constraints avoid need for checking termination for solutions

- Solutions to class $P$ of problems that satisfy constraint $C$ obtained from solutions to $P^o/C$

- $P^o/C$ can be solved using LTL (if constraints are LTL-expressible) or, in some cases, using more efficient FOND planners
Discussion

- There are many constraints that are satisfied by given target class of instances; Which constraints to make explicit?

- Can we automate the discovery of relevant constraints?

- Extend scope of QNPs that can be solved using FOND planners; General results?

- Analyze and test LTL synthesis for specific and relevant types of problems/constraints; Can existing LTL synthesis techniques be **effectively** used to solve interesting generalized planning tasks?